Evolvability Analysis: Distribution of Hyperblobs in a Variable-Length Protein Genotype Space

Hideaki Suzuki

ATR Human Information Processing Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan

Abstract

A variable-length protein genotype space (the whole amino-acid sequence space) is mathematically analyzed and a lower threshold density for adequate connectivity of functional (viable) genotypes is estimated. Functional genotypes are assumed to distribute as a 'hyperblob' which means a cluster or an island, and connectivity between hyperblobs is estimated using the theory of regular languages and the random graph theory. It is shown that the logarithmic value of the threshold density approximately decreases with an increase in the genotype length.

Introduction

Proteins are fundamental functional units of living organisms. The evolvability (the possibility of evolution of various kinds) of living things is greatly dependent upon the evolvability of proteins; hence, to enhance the evolvability of an evolutionary system, we have to augment the possible evolution of various proteins. The evolvability of protein molecules has been argued by several authors to date. About thirty years ago, Maynard-Smith (Maynard Smith, 1970) argued that 'if evolution by natural selection is to occur, functional proteins must form a continuous network which can be traversed by unit mutational steps without passing through nonfunctional intermediates.' In 1991, Lipman et al. (Lipman and Wilbur, 1991) conducted a numerical experiment using a two-dimensional conformation model of artificial proteins proposed by Lau et al. (Lau and Dill, 1989; Lau and Dill, 1990). Lipman et al. confirmed that the fictional proteins satisfy Maynard-Smith's condition. As these authors pointed out, the evolvability of proteins is largely determined by the connectivity of functional (viable) genotypes in the protein genotype space. (Here and throughout the paper, we consider the protein genotype space as the amino-acid sequence space in which points represent the amino-acid sequences of the proteins.) Since non-functional proteins quickly die out and cannot be fixed in the population, mutations, namely, slight modifications of genotypes, cannot search for various functional genotypes if the functional genotypes are sparsely distributed in the genotype space. For high evolvability, the functional proteins have to be densely distributsteps so that the functional genotypes are interconnected by unit mutational steps.

Recently, by focusing on the secondary structure of RNA molecules, Schuster et al. (Reidys *et al.*, 1997a; Reidys *et al.*, 1997b; Reidys, 1997; Schuster, 1997; Reidys *et al.*, 1998) studied the genotype space of the RNA base sequence and analyzed the connectivity of *neutral networks*. Their neutral network is a graph whose vertices represent genotypes with the same secondary structure and whose edges represent mutational changes between genotypes. They applied random graph theory to this network and derived a formula for the minimum occurrence probability of neutral mutants to ensure the high connectivity of a neutral network. However, as Schuster described in his paper (Schuster, 1997), their arguments are principally based upon the assumption that 'sequences folding into the same structure are (almost) randomly distributed in sequence space.' This prediction by inverse folding (Hofacker *et al.*, 1994) is valid for the RNA secondary structure; and yet for the conformations of protein molecules, it is expected that the amino acid sequences that create the same three-dimensional structure are distributed in an *island-like* way in the sequence space (Nishikawa, 1993). Protein genotypes with the same phenotype are likely to be unevenly distributed, so that we cannot adopt the same assumption as Shuster's for the analysis of the protein genotype space.

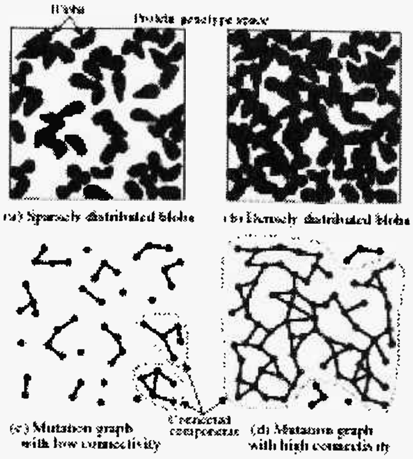
Blobs
Protein genotype space
(a) Sparsely distributed blobs
(b) Densely distributed blobs
(c) Mutation graph with low connectivity
Connected components
(d) Mutation graph with high connectivity

Figure 1: Basic concept of the genotype space analysis using blob distribution. (a) and (b) are symbolized figures of protein genotype space, and (c) and (d) are the corresponding mutation graphs. In (a) and (b), all genotypes are classified into functional ones (colored black) or nonfunctional ones (colored white). An undirected edge occurs between a node pair of the mutation graph if and only if corresponding blobs are interconnected in the genotype space. Evolvability is low for (a) and (c), and high for (b) and (d).

Based upon these notions, the author has recently analyzed the fixed-length genotype space of proteins and derived a quantitative condition for high evolvability (Suzuki, 2000a; Suzuki, 2000b). Figure 1 shows the basic concept of these papers (Suzuki, 2000a; Suzuki, 2000b). Protein genotypes with the same phenotype (function) are assumed to distribute as a *blob* which means a cluster or an island. A blob pair was regarded to be interconnected by mutation if they are adjacent or overlap in the genotype space, and, applying random graph theory (Bollobas, 1985; Palmer, 1985), the connectivity between blobs was quantitatively estimated. Although the studies reported in these papers first succeeded in formulating the minimum density of functional proteins to make a system evolvable, the studies had the following serious drawbacks. First, even if a pair of blobs are adjacent or overlap (have one common genotype at least), the size of a typical blob (here 'size' means the number of included genotypes) is so enormous that the possibility of mutation bringing about the transition from one blob to another blob is extraordinarily small. This caused overestimation of the connection (transition) probability. Second, mutational modifications causing changes in the genotype length (deletion and insertion of amino acids) were neglected, and only the substitution of amino acid bases was considered as a modification of the genotype.

The present paper remedies these problems and studies the variable-length protein genotype space. A set of genotypes with the same phenotype is referred to as a *hyperblob* or *hblob* (which includes genotypes with different lengths), and the hblob is considered to be interconnected with another by mutation if the ratio of the size of the region common to another hblob compared to the size of the hblob is larger than some constant value. (The constant value is determined from the computational resource used in the system.) Like previous studies (Suzuki, 2000a; Suzuki, 2000b), the connectivity between hblobs is represented by a graph called a *mutation graph* whose nodes correspond to hblobs and whose *directed* edges represent mutational transitions between hblobs. (Note that unlike previous studies, the edges of the mutation graph for hyperblobs are directed ones.) It is known from the random graph theory (Bollobas, 1985; Palmer, 1985) that the connectivity of a random graph dramatically changes when the ratio of the edge number to the node number passes a particular threshold value. After conducting an experiment confirming this result for a directed random graph (Section II), two numerical estimations are made for the minimum hblob number needed for high connectivity of the mutation graph (Section III). The first one actually prepares a mutation graph of randomly created hblobs and studies the growth of its connectivity. From this experiment it is shown in Section IV that there is a threshold hblob number distinguishing between a region wherein the connectivity hardly increases with the hblob number and a region wherein the connectivity swiftly increases with the hblob number. The second calculation uses a prediction by the random graph theory. The occurrence probability of a directed edge in the mutation graph is measured using a Monte Carlo estimation method, and from this probability, the threshold hblob number is calculated (Section IV). The threshold values estimated by the two methods are compared.

Preliminary Experiment

This section's purpose is to conduct a numerical experiment for a *directed* random graph and derive an experimental formula relating the threshold node number to the occurrence probability of a directed edge. Before moving on to this experiment, it will be helpful to first describe the growth of connectivity of an *undirected* random graph briefly. Let N be the order (number of vertices) of an undirected random graph and M be the number of undirected edges of the random graph. According to the random graph theory (Bollobas, 1985; Palmer, 1985), it is known that when $M < N/2$, the orders of components (connected subgraphs) of a graph are much smaller than N, whereas when $M > N/2$, a giant component is likely to emerge in the graph and the order of the largest component is comparable to N. Hence, if we express the occurrence probability of an undirected edge between a pair of nodes by P, the threshold node number N_c of an undirected random graph is given by

$$\begin{aligned} \frac{N}{2} &= M = \binom{N}{2} P = \frac{N(N-1)}{2} P \\ \therefore N_c &= \frac{1}{P} + 1 \simeq \frac{1}{P}. \text{ (undirected)} \quad (1) \end{aligned}$$

In (Suzuki, 2000b), the author numerically studied the growth of connectivity of an undirected random graph and showed that when N passes N_c given by Eq. (1), the ratio of the average number of reachable nodes from one node compared to N suddenly begins to increase and swiftly approaches one.

A similar thing happens to a *directed* random graph. Let N be the total order of a directed random graph, N_c be the threshold number of N, P be the occurrence probability of a directed edge that connects a pair of nodes in a particular direction, and e be the ratio of the average number of reachable nodes from one node by directed edges compared to N. We start the experiment with $N = 0$, add nodes one by one, generate directed edges between a new node and older nodes using P, revise a reachable node list according to the connectivity, and calculate e from the reachable node list. Numerical trials are conducted ten times using a different random number sequence for each given P value. Figure 2 shows a part of the results given for different values of P. As is clearly shown in this figure, for a directed random graph as well, e suddenly begins to increase after N passes a particular threshold value N_c. N_c is determined experimentally as

$$N_c \simeq \frac{1.15}{P}, \qquad \text{(directed)} \qquad (2)$$

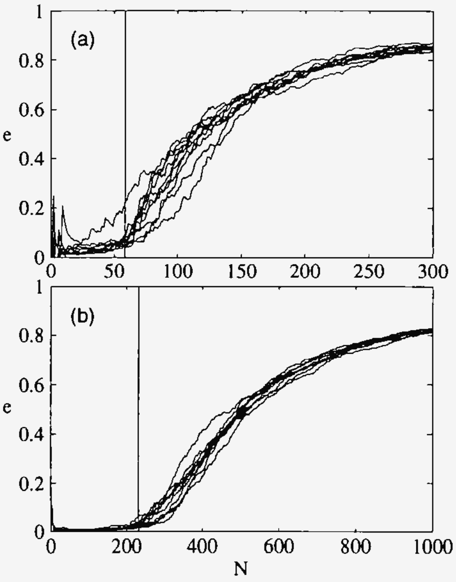

Figure 2: The growth of e as a function of N for a directed random graph with (a) $P = 0.02$ or (b) $P = 0.005$. The straight vertical lines represent N_c values given by Eq. (2).

Methods to Analyze Evolvability

Hyperblobs Represented by Regular Expressions

In this paper, the author represents a protein genotype by a sequence (string) of I characters chosen among K different characters (K is a fixed number and I is a variable). For a natural protein, an character can be compared to an amino acid base (I is about several hundred and $K = 20$), and for a subroutine of a machine-language programming system like Tierra (Ray, 1992; Ray, 1997; Ray and Hart, 1998), an character can be compared to an instruction (I is about several dozen and $K = 32$). In both example systems, the genotype of a protein/subroutine includes functionally important bases/instructions and functionally unimportant bases/instructions. The substitution or deletion of functionally important bases/instructions crucially changes the entire function of the molecule/subroutine, whereas the substitution, deletion, or even insertion of functionally unimportant bases/instructions usually has no influence on the functionality of the entire molecule/subroutine. Considering this characteristic, here we represent a hyperblob (a set of neighboring genotypes with the same function) by a sequence of I_{u} 'unsubstitutable' (functionally important) characters including insertions of arbitrary numbers of 'substitutable' (functionally unimportant) characters. I_{s} sequences are allowed to be inserted between some choice of positions between the unsubstitutable characters, and for each inserted sequence, characters are chosen out of K_{s} substitutable characters. Here the parameter ranges are $0 \leq I_{\mathrm{u}} \leq I$, $0 \leq I_{\mathrm{s}} \leq I_{\mathrm{u}}+1$, $0 \leq K_{\mathrm{s}} \leq K$.

Such a set of sequences is simply represented by a kind of regular expression used in the theory of formal languages (Hopcroft and Ullman, 1969; Hopcroft and Ullman, 1979). Let, for example, an character set be represented by $\{\alpha, \beta, \gamma, \delta\}$ ($K = 4$). A regular expression for an hblob on this character set is a formula like

$$\alpha \cdot (\alpha + \gamma)^{*} \cdot \beta\alpha \cdot (\beta + \gamma + \delta)^{*}, \qquad (3)$$

where '$\cdot$' represents *concatenation* ('$\cdot$' is often omitted), '$+$' represents *union* (or *or*), and

$$\mathrm{x}^{*} = 1 + \mathrm{x} + \mathrm{xx} + \mathrm{xxx} + \cdots$$

is a *closure* operation (1 represents a *null* string). Eq. (3) can be considered to represent an hblob by regarding the terms α and $\beta\alpha$ as unsubstitutable characters and the closure operations $(\alpha + \gamma)^{*}$ and $(\beta + \gamma + \delta)^{*}$ as the insertion of substitutable characters. In this example, the hblob parameter values are $I_{\mathrm{u}} = 3$, $I_{\mathrm{s}} = 2$, and $K_{\mathrm{s}} = 2$ or 3.

The size of (the number of genotypes belonging to) hblob $S(I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}})$ is formulated as follows. As shown in Fig 3, the minimum length of the genotypes of an hblob is I_{u}, whereas the maximum length of the genotypes of an hblob is infinite. So, in this paper, we limit the length of genotypes to $I_{\max}$ and express $S(I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}})$ by the sum of section areas at length-I subspaces as

$$S(I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}}) = \sum_{I=I_{\mathrm{u}}}^{I_{\max}} s(I; I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}}) \qquad (4)$$

To formulate $s(I; I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}})$, here the author adopts the assumption that the values of K_{s} in a regular expression are the same (hereafter, this is assumed throughout the paper). Then $s(I; I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}})$ is approximately given by .

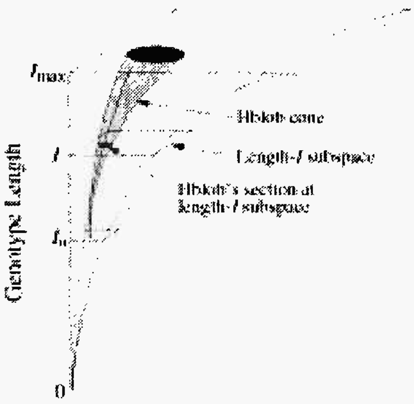

Figure 3: Symbolized picture of a hyperblob in the variable-length protein genotype space. The size of the length-I subspace increases exponentially with I. An hblob covers a cone-like region whose apex is located in the length-I_{u} subspace and whose base can extend freely toward large I values.

$$s(I; I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}}) \simeq \binom{I - I_{\mathrm{u}} + I_{\mathrm{s}} - 1}{I_{\mathrm{s}} - 1} \cdot {K_{\mathrm{s}}}^{I - I_{\mathrm{u}}}. \qquad (5)$$

See Appendix A for the detailed derivation.

Transition Probability between Hyperblobs

The analyses in the subsequent sections are based upon the following assumptions:

- Hblobs are uniformly distributed in the genotype space, allowing a pair of hblobs to overlap (have common genotypes). (Although no two hblobs can overlap actually, we here allow this possibility by assuming a random distribution of hblobs.)

- A population of protein genotypes in an hblob visits all inner genotypes uniformly, so that the transition probability from one hblob to another is calculated from the ratio of the size of the overlapped region compared to the size of the hblob.

- Although the transition from hblob-(a) to hblob-(b) actually happens if a set of mutant genotypes created from hblob-(a) overlaps with hblob-(b), we assumes that the transition happens only if hblob-(a) overlaps with hblob-(b).

Figure 4 symbolically shows the definition of the transition probability between a pair of hblobs. Using the last two assumptions, the transition probability from hblob-(a) to hblob-(b) is

$$r_{\rm a \rightarrow b} \equiv \frac{S({\rm a} \wedge {\rm b})}{S({\rm a})}. \qquad (6)$$

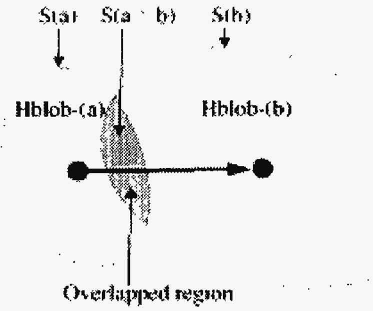

Figure 4: Symbolized figure of hblobs and corresponding mutation graph. S(a), S(b), and S(a∧b) are the size of hblob-(a), hblob-(b), and the overlapped region, respectively. In this figure, a directed edge from hblob-(a) to hblob-(b) occurs because the transition probability $r_{\rm a\rightarrow b}$ satisfies $r_{\rm a\rightarrow b} = S({\rm a} \wedge {\rm b})/S({\rm a}) > r_0$, whereas a directed edge from hblob-(b) to hblob-(a) does not occur because $r_{\rm b\rightarrow a} = S({\rm a} \wedge {\rm b})/S({\rm b}) < r_0$.

To calculate $S({\rm a} \wedge {\rm b})$, here we use the regular expression for an hblob and the nondeterministic finite automaton (NFA) which accepts the regular expression (Hopcroft and Ullman, 1969; Hopcroft and Ullman, 1979). If we are given two regular expressions for a pair of hblobs, we can make two NFAs that accept those expressions, combine those NFAs into one NFA, and derive a regular expression accepted by the combined NFA. The obtained regular expression represents genotypes common with the two original regular expressions. The author illustrates this procedure in the following.

Let the character set be $\{\alpha, \beta, \gamma, \delta\}$ ($K = 4$) and a pair of hblobs (hblob-(a) and hblob-(b)) be represented by regular expressions

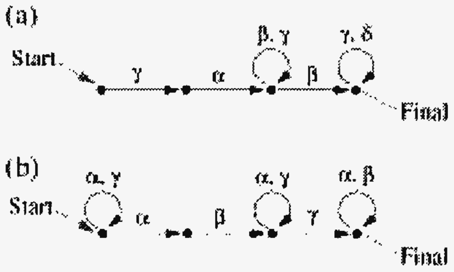

Figure 5: (a) NFA that accepts Expr. (7a), and (b) NFA that accepts Expr. (7b). Black dots represent states, straight arrows correspond to unsubstitutable characters, and loops correspond to closure operations (substitutable characters).

$$\gamma \cdot \alpha \cdot (\beta + \gamma)^* \cdot \beta \cdot (\gamma + \delta)^*, \quad (7a)$$

$$(\alpha + \gamma)^* \cdot \alpha \cdot \beta \cdot (\alpha + \gamma)^* \cdot \gamma \cdot (\alpha + \beta)^*. \quad (7b)$$

The parameter values for these hblobs are $I_{\rm u}^{(\rm a)} = 3$, $I_{\rm s}^{(\rm a)} = 2$, and $K_{\rm s}^{(\rm a)} = 2$ for hblob-(a) and $I_{\rm u}^{(\rm b)} = 3$, $I_{\rm s}^{(\rm b)} = 3$, and $K_{\rm s}^{(\rm b)} = 2$ for hblob-(b). NFAs that accept these expressions are shown in Fig. 5, and the schematic figures of these NFAs and the combined NFA are shown in Fig. 6. A string accepted by the NFA in Fig. 6(c) is accepted by both original NFAs, so that genotypes included in the overlapped (common) region of the two original hblobs is represented by regular expressions accepted by the NFA in Fig. 6(c), or in other words, the paths that begin at the starting state and end at the final state of Fig. 6(c). In the present example, there are two paths represented by the regular expression

$$\gamma \cdot \alpha \cdot \beta \cdot \gamma^* \cdot \gamma + \gamma \cdot \alpha \cdot \beta \cdot \gamma^* \cdot \gamma \cdot \beta^* \cdot \beta. \quad (8)$$

Here, the parameter values for the first term are $I_{\rm u} = 4$, $I_{\rm s} = 1$, and $K_{\rm s} = 1$, and, for the second term, $I_{\rm u} = 5$, $I_{\rm s} = 2$, and $K_{\rm s} = 1$. Generally speaking, the values of $K_{\rm s}$ are different for closure operations in a regular expression; and yet for simplicity, here we assume that $K_{\rm s}$'s in a common regular expression have a flat value equal to an integer nearest to $K_{\rm s}^{(\rm a)} K_{\rm s}^{(\rm b)}/K$ (which is one for the present example). (We also apply this assumption to all

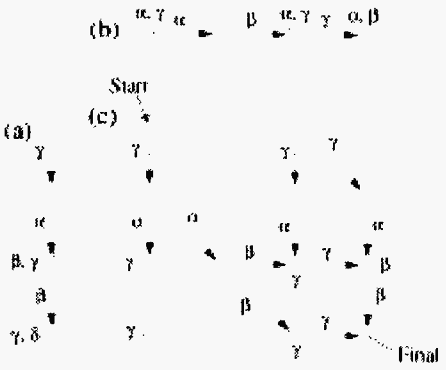

Figure 6: Schematic figures of NFAs that accept (a) Expr. (7a), (b) Expr. (7b), and (c) strings common with Exprs. (7a) and (7b). Black dots for states are omitted here, and loops are represented by white circles. (c) is created on the two dimensional grid whose vertical lines correspond to arrows of (a) and whose horizontal lines correspond to arrows of (b). In (c), a vertical arrow occurs if and only if the corresponding state of (b) has a loop whose character set includes the character of (a), a horizontal arrow occurs if and only if the corresponding state of (a) has a loop whose character set includes the character of (b), a loop occurs if and only if the corresponding states of (a) and (b) have loops whose character sets have common elements, and an oblique arrow occurs if and only if the corresponding arrows of (a) and (b) have the same character.

subsequent analyses.) Finally the transition probability $r_{\mathrm{a}\to\mathrm{b}}$ is derived as

$$r_{\mathrm{a}\to\mathrm{b}} = \frac{S(4,1,1)+S(5,2,1)}{S(3,2,2)}. \qquad (9)$$

Strictly speaking, there is a possibility that regular expressions corresponding to different paths in the combined NFA have common strings; however we approximately neglected this possibility and calculated the numerator of Eq. (9) by the sum of the sizes of the common regular expressions. This approximation is also applied to all subsequent calculations. Eq. (9) can be evaluated for a fixed value $l_{\max}$ by substituting Eqs. (4) and (5).

Directed Mutation Graph

A directed mutation graph is a graph whose nodes correspond to hblobs and whose directed edges represent mutational transitions between hblobs (Fig. 4). A directed edge connecting a pair of hblobs occurs if and only if the transition probability from the 'initial' hblob to the 'terminal' hblob r is larger than a particular constant value r_0. r_0 is determined so that if $r > r_0$, a population of genotypes distributed in an hblob might be able to find a common region with another hblob in a practical waiting time. (For natural proteins which have an enormous number of individuals in each generation, the value of r_0 is extremely small, but for an ALife system that is run in a computer, the r_0 value is taken to be rather large depending upon the computational resources used in the experiment.)

The evolvability of proteins is measured by the connectivity of the mutation graph. Let the 'evolvability ratio' e be defined as the ratio of the average number of reachable nodes from one node compared to the total node number of the mutation graph. When e is large and comparable to one ($e \sim 1$), there is a strong possibility that evolution starting with one genotype will explore through the whole genotype space by mutational transitions between hblobs and create a variety of genotypes in time. This makes proteins highly evolvable. When e is much smaller than one ($e \ll 1$), on the other hand, most hblobs are isolated and evolution starting at one genotype is very likely to be confined to the initial hblob or its neighborhood and mutation cannot explore through possible hblobs in the genotype space. This makes proteins less evolvable.

As was shown in the preliminary experiment, if a directed edge in the mutation graph occurs randomly according to the occurrence probability $P \equiv \mathrm{Prob}(r > r_0)$, it is expected that e of the mutation graph will begin to increase drastically with the hblob number N after N exceeds some threshold value N_c. The author evaluates this threshold value with two different methods in the following section.

Direct Method

The first method is a direct one that creates a real example of the mutation graph of generated hblobs. In this method, regular expressions representing hblobs are created one by one using a random number sequence, the connectivity between a newly created hblob and previous hblobs is checked using the method of NFAs, the mutation graph is created according to the connectivity, and the evolvability ratio e is calculated from the mutation graph.

The hblob size S is determined by the hblob parameters l_{u}, l_{s}, K_{s} and the maximum genotype length $l_{\max}$ (Eqs. (4) and (5)). In the experiment, I_{max} is fixed to a particular constant number, and the values for I_{u}, I_{s}, and K_{s} are variously determined using Beta distributions. Beta distribution is a convenient probability distribution function on the range [0,1] whose mean and variance are freely chosen by adjusting parameters.

$$p_{\mathrm{min}}(I_{\mathrm{u}}) \propto f_{\mathrm{BET}}\left(\frac{I_{\mathrm{u}}}{I_{\mathrm{max}}}; \mu_{\mathrm{min}}, \sigma_{\mathrm{min}}\right) \quad (10a)$$

$$p_{\mathrm{ins}}(I_{\mathrm{s}}) \propto f_{\mathrm{BET}}\left(\frac{I_{\mathrm{s}}}{I_{\mathrm{u}}+1}; \mu_{\mathrm{ins}}, \sigma_{\mathrm{ins}}\right) \quad (10b)$$

$$p_{\mathrm{sub}}(K_{\mathrm{s}}) \propto f_{\mathrm{BET}}\left(\frac{K_{\mathrm{s}}}{K}; \mu_{\mathrm{sub}}, \sigma_{\mathrm{sub}}\right). \quad (10c)$$

$p_{\ldots}(X)$s are normalized so as to satisfy $\sum_X p_{\ldots}(X) = 1$ and f_{BET} is a Beta distribution given by

$$f_{\mathrm{BET}}(x;\mu,\sigma) = \frac{x^{v-1}(1-x)^{w-1}}{B(v,w)}, \quad (11a)$$

$$v = \frac{\mu^2-\mu^3}{\sigma^2} - \mu, \quad (11b)$$

$$w = \left(\frac{1}{\mu}-1\right)\left(\frac{\mu^2-\mu^3}{\sigma^2}-\mu\right). \quad (11c)$$

v and w are determined so that μ and σ might be the average and the standard deviation of a Beta distribution, respectively. Throughout this paper, we use $\mu_{\mathrm{min}} = 0.4$, $\sigma_{\mathrm{min}} = 0.07$, $\mu_{\mathrm{ins}} = 0.7$, $\sigma_{\mathrm{ins}} = 0.05$, $\mu_{\mathrm{sub}} = 0.7$, and $\sigma_{\mathrm{sub}} = 0.05$ (Fig. 7). μ_{ins} and μ_{sub} are taken to be larger than μ_{min} in order to consider a situation which allows the insertion of a fairly larger number of redundant characters than functionally important characters.

Numerical trials are conducted twenty times using different random number sequences, and for each trial, e is calculated as a function of N. The simulation is conducted on a Linux computer with Pentium II processor (333MHz) and 128MB main memory.

Monte Carlo Method

The second method calculates the edge's occurrence probability P by the Monte Carlo method and estimates N_c from theoretical formula Eq. (2) directly. Here we again assume that the hblob-size parameters I_{u}, I_{s}, and K_{s} obey the Beta distributions in Eqs. (10a)~(10c). Based upon this distribution, the expected value of P is a function of I_{max}, K, μ_{min}, σ_{min}, μ_{ins}, σ_{ins}, μ_{sub}, σ_{sub}, and r_0 and formulated as

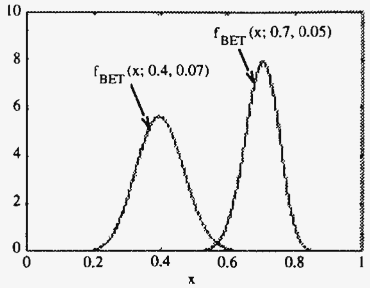

Figure 7: Beta distributions given by Eqs. (11a)~(11c) substituted with $\mu = 0.4$ and $\sigma = 0.07$ or $\mu = 0.7$ and $\sigma = 0.05$.

$$\begin{aligned}&\overline{P}(I_{\mathrm{max}}, K, \mu_{\mathrm{min}}, \sigma_{\mathrm{min}}, \mu_{\mathrm{ins}}, \sigma_{\mathrm{ins}}, \mu_{\mathrm{sub}}, \sigma_{\mathrm{sub}}, r_0)\\ &= \sum_{I_{\mathrm{u}}^{(\mathrm{a})}} \cdots \sum_{K_{\mathrm{s}}^{(\mathrm{b})}} P(I_{\mathrm{u}}^{(\mathrm{a})}, I_{\mathrm{u}}^{(\mathrm{b})}, I_{\mathrm{s}}^{(\mathrm{a})}, I_{\mathrm{s}}^{(\mathrm{b})}, K_{\mathrm{s}}^{(\mathrm{a})}\\ &\quad , K_{\mathrm{s}}^{(\mathrm{b})}) \times p_{\mathrm{min}}(I_{\mathrm{u}}^{(\mathrm{a})}) p_{\mathrm{min}}(I_{\mathrm{u}}^{(\mathrm{b})}) p_{\mathrm{ins}}(I_{\mathrm{s}}^{(\mathrm{a})})\\ &\quad \times p_{\mathrm{ins}}(I_{\mathrm{s}}^{(\mathrm{b})}) p_{\mathrm{sub}}(K_{\mathrm{s}}^{(\mathrm{a})}) p_{\mathrm{sub}}(K_{\mathrm{s}}^{(\mathrm{b})}). \quad (12)\end{aligned}$$

To calculate $P(I_{\mathrm{u}}^{(\mathrm{a})}, \cdots, K_{\mathrm{s}}^{(\mathrm{b})})$, we randomly generate a large number of pairs of hblob-(a) and hblob-(b), check if the transition probabilities between pairs ($r_{\mathrm{a}\to\mathrm{b}}$s) are larger than r_0 or not one by one, and calculate $P = \mathrm{Prob}(r > r_0)$ statistically. The trial number of pairs is chosen so that at least five pairs of hblobs might satisfy $r > r_0$ or the number might not exceed one million. Since this calculation procedure is awfully time-consuming, the author adopts the following approximations to make an estimation within a practical waiting time. First, the author limits the number of paths in the combined NFA to 500,000 and discards paths exceeding this number. Second, the summation of Eq. (12) is carried out only for the terms satisfying

$$p_{\mathrm{min}}(I_{\mathrm{u}}^{(\mathrm{a})}) \times \cdots \times p_{\mathrm{sub}}(K_{\mathrm{s}}^{(\mathrm{b})}) > p_{\mathrm{th}}$$

using p_{th} determined by the condition that the partial sum of $p_{\mathrm{min}}(I_{\mathrm{u}}^{(\mathrm{a})}) \times \cdots \times p_{\mathrm{sub}}(K_{\mathrm{s}}^{(\mathrm{b})})$ exceeds 0.95. In addition, the terms satisfying $0.8(I_{\mathrm{u}}^{(\mathrm{a})} + I_{\mathrm{u}}^{(\mathrm{b})}) \geq I_{\mathrm{max}}$ are also omitted from the summation because for these terms, the length (I_{u} values) of paths in the combined NFA is very likely to exceed I_{max}. The calculated $\overline{P}$ is renormalized by the partial sum of $p_{\text{min}}(I_u^{(a)}) \times \cdots \times p_{\text{sub}}(K_s^{(b)})$. The evaluation program was written in C language and run on the same computer as that used in the experimental method.

Density of Functional Genotypes

In this subsection, the author describes the relation between the hblob number N and the density of functional protein genotypes ρ. If we neglect the overlapped regions between hblobs, a N value can be transformed to a ρ value using the average size of hblob $\overline{S}$ as

$$\begin{aligned}\rho &= \frac{N \times \overline{S}}{\text{Total number of genotypes}} \\ &= \frac{N \times \overline{S}}{\sum_{I=0}^{I_{\text{max}}} K^I} \\ &= N \times \overline{S} \times \frac{K-1}{K^{I_{\text{max}}+1}-1}. \qquad (13)\end{aligned}$$

Like $\overline{P}$, $\overline{S}$ is calculated by

$$\begin{aligned}&\overline{S}(I_{\text{max}}, K, \mu_{\text{min}}, \sigma_{\text{min}}, \mu_{\text{ins}}, \sigma_{\text{ins}}, \mu_{\text{sub}}, \sigma_{\text{sub}}) \\ &\quad = \sum_{I_u}\sum_{I_s}\sum_{K_s} S(I_u, I_s, K_s) \cdot p_{\text{min}}(I_u) \\ &\qquad \times p_{\text{ins}}(I_s) p_{\text{sub}}(K_s). \qquad (14)\end{aligned}$$

Eq. (14) is evaluated numerically using Eqs. (4), (5), and (10a)~(10c). Eq. (13) is used for relating the threshold number N_c to the threshold density of functional proteins ρ_c.

Results

Figure 8 shows the results of the direct method given from a three-day simulation run. From this figure, we can conclude that the evolvability ratio e increases with the hblob number N. Compared to Fig. 2, it is harder to observe the existence of the threshold number N_c in these figures; and yet there certainly exists a positive number N_c below which e hardly increases with N and above which e increases fairly swiftly with N. These values are $N_c \sim 200$ for (a) and $N_c \sim 750$ for (b).

When the number of hblobs is smaller than these threshold values ($N < N_c$), e is very likely to be near zero and mutation cannot easily cause changes from one hblob to another, resulting in limited protein evolvability. When the number of hblobs is larger than this value ($N > N_c$), on the other hand, the number of hblobs reachable from one functional genotype swiftly increases with the number of hblobs. When this happens, mutation can explore very widely through the genotype space and create a variety of functional protein genotypes, resulting in high protein evolvability. Accordingly, N_c can be regarded as the threshold value used for evaluating the extent of evolvability.

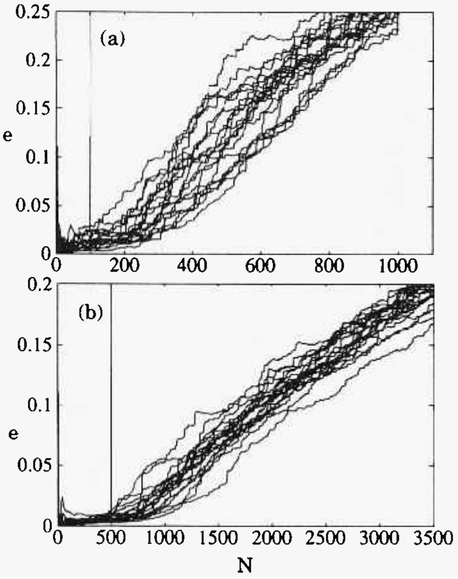

Figure 8: Evolvability ratio e as a function of hblob number N given by the direct method. Parameter values are $I_{\text{max}} = 20$, $K = 10$, $\mu_{\text{min}} = 0.4$, $\sigma_{\text{min}} = 0.07$, $\mu_{\text{ins}} = 0.7$, $\sigma_{\text{ins}} = 0.05$, $\mu_{\text{sub}} = 0.7$, $\sigma_{\text{sub}} = 0.05$, and (a) $r_0 = 0.0001$ or (b) $r_0 = 0.0003$. The vertical straight lines show the theoretical estimation of N_c given from the extrapolation in Fig. 9(a).

The reason why e does not increase drastically above the threshold values is inferred as follows. In the present experiment, the hblob parameters I_u, I_s, and K_s are chosen from the Beta distributions, and the sizes of created hblobs are diversely distributed. Because the occurrence probabilities of directed edges are strongly dependent upon the hblob sizes (Fig. 4), the difference in hblob sizes causes an uneven distribution of the occurrence probabilities of directed edges. This is considered to hinder e from increasing drastically in the region $N > N_c$.

Figure 9 shows the results of the Monte Carlo method given by a five-day simulation run. Because the Monte Carlo trial number increases extraordinarily with an increase in

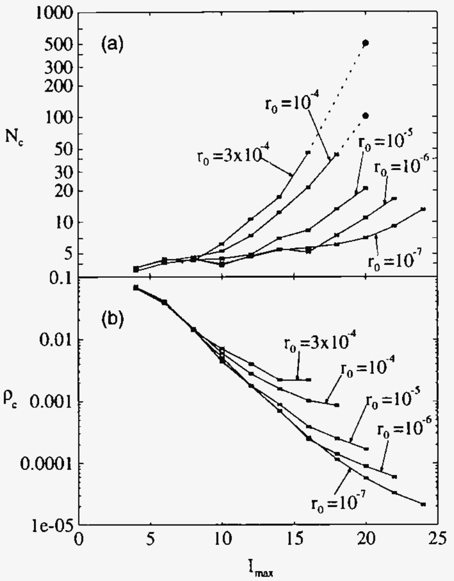

Figure 9: (a) Threshold hblob number N_c and (b)threshold density of functional genotypes ρ_c as a function of maximum genotype length I_{max} under different values of r_0. The other parameter values are $K = 10$, $\mu_{min} = 0.4$, $\sigma_{min} = 0.07$, $\mu_{ins} = 0.7$, $\sigma_{ins} = 0.05$, $\mu_{sub} = 0.7$, and $\sigma_{sub} = 0.05$. The dotted lines and black circles show the extrapolation for the values at $I_{max} = 20$. (b) was calculated from (a) using Eq. (13).

I_{max}, the results are given only for the limited ranges of I_{max}. We can say from Fig. 9(a) that $\log N_c$ approximately increases linearly with I_{max} in the regions of larger I_{max}. By extrapolating from this dependence, the author estimated the N_c values for $I_{max} = 20$, which are shown in Fig. 8 by vertical straight lines. Although the agreement between the two evaluation methods is not a precise one, the Monte Carlo method is considered to succeed in making a rough estimation of N_c in spite of a number of assumptions and approximations in the calculation.

Figure 9(b) says that for a fixed r_0, $\log \rho_c$ approximately decreases linearly with I_{max} although in the regions of larger I_{max}, the values deviate from this law depending upon r_0.

Discussion

To quantitatively estimate the evolvability of natural or artificial proteins, a hyperblob representing neighboring genotypes with the same function was introduced and its connectivity was studied. A set of variable-length genotypes included in an hblob was expressed using the regular expression, and a method was established to estimate the connectivity between a pair of hblobs using the theory of the regular languages and automata. The connectivity of all hblobs was represented by a directed mutation graph, and by evaluating the connectivity, the minimum number of hyperblobs for high connectivity was calculated. It was concluded that the logarithmic value of the minimum density of functional genotypes for high evolvability approximately decreases with the maximum length of a protein genotype.

Evolvability has been one of the most widely studied topics in the study of artificial life. Bedau et al. (Bedau and Packard, 1992; Bedau *et al.*, 1998; Rechtseiner and Bedau, 1999) have proposed using the neutral shadow model of a target system as a no-adaptation null hypothesis; Ray et al. (Standish, 1999; Ray and Xu, 2000) have studied several evolvability measures in Tierra; and the author and Ray (Suzuki and Ray, 2000) have proposed a set of design criteria to enhance the evolvability of an ALife system using a target system named SeMar (Suzuki, 1998; Suzuki, 1999; Suzuki, 2000c). Among them, the author's approach (Suzuki, 2000a; Suzuki, 2000b) has a unique characteristic in that it tries to study the distribution of functional genotypes in the *whole* protein genotype space. Although the results given in this paper (Fig. 9) cover only a limited region of parameter values, if more numerical experiments are conducted using the parameter values tailored for proteins (that is, functional units) of an ALife system, a derived ρ_c value might be used as a threshold value to discern whether or not the system has high evolvability. In order to make an ALife system highly evolvable, the system design must be optimized so that the ρ value (which can be estimated statistically) might be greater than ρ_c (Suzuki and Ray, 2000).

Appendix A: Derivation of Eq. (5)

$s(I; I_u, I_s, K_s)$ is the number of different strings with length I generated by the expansion of closure operations in a regular expression. That is approximately given by the product of the number of different index sets for the closure operations (represented by the repeated combination) and the number of different substitutable character sets in a string;

$$\begin{aligned} s(I; I_{\mathrm{u}}, I_{\mathrm{s}}, K_{\mathrm{s}}) &\simeq {}_{I_{\mathrm{s}}}H_{I-I_{\mathrm{u}}} \cdot {K_{\mathrm{s}}}^{I-I_{\mathrm{u}}} \\ &= \binom{I - I_{\mathrm{u}} + I_{\mathrm{s}} - 1}{I_{\mathrm{s}} - 1} \cdot {K_{\mathrm{s}}}^{I-I_{\mathrm{u}}} \end{aligned}$$

Note that there is a possibility that the right-hand side of this equation might be an overestimation on account of the repeated counting of the same string. For example, although the expansion of $(\alpha + \beta)^* \cdot \alpha \cdot (\alpha + \beta)^*$ generates only 3 strings with length two ($\alpha\alpha$, $\alpha\beta$, and $\beta\alpha$), Eq. (5) gives $s(2; 1, 2, 2) = \binom{2}{1} \times 2 = 4$. This is because the string $\alpha\alpha$ is counted twice by distinguishing '$1 \cdot \alpha \cdot \alpha$' and '$\alpha \cdot \alpha \cdot 1$'.

References

Bedau, M.A., Packard, N.H.: Measurement of evolutionary activity, teleology, and life. In Proc. of *Artificial Life II* (1992) 431-461

Bedau, M.A., Snyder, E., Packard, N.H.: A Classification of Long-Term Evolutionary Dynamics. In Proc. of *Artificial Life VI* (1998) 228-237

Bollobás, B.: Random graphs. Academic Press, London (1985)

Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie* **125** (1994) 167-188

Hopcroft, J.E., Ullman, J.D.: Formal Languages and Their Relation to Automata. Addison-Wesley, Massachusetts (1969)

Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Massachusetts (1979)

Lau, F.L., Dill, K.A.: A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules* **22** (1989) 3986–3997

Lau, F.L., Dill, K.A.: Theory for protein mutability and biogenesis. *Proc. Natn. Acad. Sci. U.S.A.* **87** (1990) 638–642

Lipman, D.J., Wilbur, W.J.: Modelling neutral and selective evolution of protein folding. *Proc. R. Soc. Lond. B* **245** (1991) 7-11

Maynard Smith, J.: Natural selection and the concept of a protein space. *Nature, Lond.* **225** (1970) 563–564

Nishikawa, K.: Island hypothesis: Protein distribution in the sequence space. *Viva Origino* **21** (1993) 91-102

Palmer, E.M.: Graphical evolution. John Wiley & Sons, New York (1985)

Ray, T.S.: An approach to the synthesis of life. In Proc. of *Artificial Life II* (1992) 371–408

Ray, T.S.: Selecting Naturally for Differentiation. In Proc. of *Genetic Programming* (1997) 414–419

Ray, T.S., Hart, J.: Evolution of differentiated multi-threaded digital organisms. In Proc. of *Artificial Life VI* (1998) 295–304

Ray, T.S., Xu, C.: Measures of evolvability in tierra. In Proc. of *Artificial Life and Robotics* Vol. 1 (2000) I-12-I-15

Rechtseiner, A., Bedau, M.A.: A genetic neutral model for quantitative comparison of genotypic evolutionary activity. In Proc. of *European Conference on Artificial Life* (1999) 109-118

Reidys, C., Stadler, P.F., Schuster, P.: Genetic properties of combinatory maps - Neutral networks of RNA secondary structures. *Bull. Math. Biol.* **59** (1997) 339–397 or Santa Fe Working Paper #95-07-058 available at http://www.santafe.edu/sfi/publications/working-papers.html

Reidys, C., Kopp, S., Schuster, P.: Evolutionary optimization of biopolymers and sequence structure maps. In Proc. of *Artificial Life V* (1997) 379-386

Reidys, C.: Random induced subgraphs of generalized n-cubes. *Adv. in Appl. Math.* **19** (1997) 360–377

Reidys, C., Forst, C.V., Schuster, P.: Replication and mutation on neutral networks. Submitted to *Bull. of Math. Biol.*, or Santa Fe Working Paper #98-04-036 available at http://www.santafe.edu/sfi/publications/working-papers.html

Schuster, P.: Landscapes and molecular evolution. *Physica D* **107** (1997) 351–365 or Santa Fe Working Paper #96-07-047 available at http://www.santafe.edu/sfi/publications/working-papers.html

Standish, R.K.: Some techniques for the measurement of complexity in Tierra. In Proc. of *European Conference on Artificial Life* (1999) 104-108

Suzuki, H.: One-dimensional unicellular creatures evolved with genetic algorithms. In Proc. of *Joint Conference on Information Sciences* Vol. II (1998) 411–414

Suzuki, H.: A Simulation of Life Using a Dynamic Core Memory Partitioned by Membrane Data. In Proc. of *European Conference on Artificial Life* (1999) 412-416

Suzuki, H.: Minimum Density of Functional Proteins to Make a System Evolvable. In Proc. of *Artificial Life and Robotics* Vol. 1 (2000) 30-33

Suzuki, H., Ray, T.S.: Conditions to Facilitate the Evolvability of Digital Proteins. In Proc. of *Joint Conference on Information Sciences* Vol. I (2000) 1078–1082

Suzuki, H.: Evolvability Analysis Using Random Graph Theory. To be published in Proc. of *Asian Fuzzy Systems Symposium* (2000)

Suzuki, H.: An Approach to Biological Computation: Unicellular Core-Memory Creatures Evolved Using Genetic Algorithms. *Artificial Life* **5** (2000) 367–386